

Booléens

Il s'agit d'un type particulier car il ne possède que deux valeurs possibles: **True**, **False**.

Les booléens permettent d'indiquer si une assertion (que l'on peut construire à l'aide de différentes relations de comparaison) est vraie ou non.

Voici les opérateurs de comparaison:

Relation mathématique	Instruction Python
égalité	<code>==</code>
différent (\neq)	<code>!=</code>
inférieur strict	<code><</code>
supérieur strict	<code>></code>
inférieur ou égal	<code><=</code>
supérieur ou égal	<code>>=</code>

Exemple:

```
2>3
3.1>=2.5
2=3
2==3
(1+1)!=2
```

Remarque 1 ATTENTION! `==` évalue une condition et ne doit pas être confondue avec `=`

Opérations sur les booléens:

Opérateur logique	Instruction Python
conjonction <i>et</i>	<code>and</code>
disjonction <i>ou</i>	<code>or</code>
négation <i>non</i>	<code>not</code>

Exemple:

```
not (3>2)
not (1>2)
```

Remarque 2 :

La négation d'une assertion ne fournit pas l'écriture de l'assertion contraire. On obtient seulement la valeur booléenne contraire de l'assertion de départ.

Autrement dit, la réponse de Python à la commande `not (3>2)` n'est pas la commande `(3<=2)` mais la valeur booléenne de cette dernière, à savoir **False**.

Remarque 3 On peut enchaîner les comparaisons: `x<y>z!=t` signifie `x<y and y>z and z != t`.

On évitera de trop enchaîner tout de même, sous peine de ne plus rien y comprendre...!

Exemple:

```
3.5!=3<5
1.5<2.5<2
```